Time Solutions

Copyright © James Raynard 2021

Clocks

- What are the main differences between system_clock and steady_clock?
  - system_clock is based on the hardware clock in the system. It represents the time as displayed by analogue clocks
  - The time as measured by system_clock may change erratically, due to daylight saving changes, clock synchronization, leap seconds, etc
  - steady_clock is an idealized clock
  - The time as measured by steady_clock only increases one tick at a time. It can never jump or go backwards

Copyright © James Raynard 2021

Clocks

- Give an example of where each one should be used
  - system_clock gives the time as perceived by human users and other systems
  - It should be used when interfacing to humans (GUIs) or other systems
  - steady_clock will always give the correct result when measuring intervals between events
  - It should be used when we need a "stopwatch" facility

Copyright © James Raynard 2021

Duration units

- Write down expressions that define duration objects which represent intervals of
  - 2 seconds
  - 20 milliseconds
  - 50 microseconds

```
seconds(2)
milliseconds(20)
microseconds(50)
```

Copyright © James Raynard 2021

std::chrono literals

- Repeat the previous exercise, using standard literals

```
2s
20ms
50us
```

Copyright © James Raynard 2021

Useful functions for working with time

- Write down an expression which returns the current time from the hardware clock

  system_clock::now()

- Write a program which prints out a message, pauses for a certain time interval, then prints out another message

Copyright © James Raynard 2021

Time point

- Write a program which prints out the time taken to perform a function call

Copyright © James Raynard 2021